NC Edit 9.0

Benutzerhandbuch

Inhaltsverzeichnis

1. EINLEITUNG 3

2. HAFTUNGSAUSSCHLUSS 4

4. INSTALLATION 5

4.1 EINGABE DER AUTORISIERUNG 5
4.2 NETZWERKINSTALLATION 5

5. MENÜS UND BEFEHLE 7

5.1 DAS HAUPTFENSTER 7
5.2 DAS MENÜ DATEI 8
5.3 DAS MENÜ BEARBEITEN 9
5.4 DAS MENÜ SUCHEN 11
5.5 DAS MENÜ ÜBERTRAGEN 13
5.6 DAS MENÜ MAKROS 15
5.7 DAS MENÜ OPTIONEN 15
5.7 DAS MENÜ FENSTER 17
5.8 DAS MENÜ HILFE 17

6. KONFIGURATION 18

6.1 KARTEIKARTE ANPASSEN 19
6.2 AUTOMATISCHES SICHERN, BACKUPS 20
6.3 STATUS SICHERN 21

7. REGULÄRE AUSDRÜCKE 22

7.1 BEISPIEL: SUCHEN NACH KLAMMERAUSDRÜCKEN 23
7.2 BEISPIEL: SUCHEN NACH ANFÜHRUNGSZEICHEN 23
7.3 BEISPIEL: LEERZEILEN SUCHEN UND LÖSCHEN 24
7.4 BEISPIEL PARAMETERTAUSCH 24
8.1 TASTATURMAKROS 25
8.2 MAKROS MIT DEM COMPILER ERSTELLEN 26
8.2.1 STRUKTUR DER COMPILERSPRACHE 28
8.2.2 BEFEHLSREFERENZ DES COMPILERS 29
8.2.2.1 Block Operationen *29*
8.2.2.2 Cursorpositionierung *29*
8.2.2.3 Stringmanipulationen *30*

1. Einleitung

In diesem Handbuch finden Sie Informationen über die allgemeine Bedienung, die Anpassungsmöglichkeiten und die Makroprogrammierung. In den ersten Kapiteln finden Sie alle Menüpunkte erläutert. Dieser Abschnitt ist kurz gehalten, da die Bedienung eines Windows Editors trivial ist.

Die Anpassungsmöglichkeiten sind ausführlicher dargelegt. Der NC Edit bietet umfangreiche Möglichkeiten die Oberfläche, die Menüs und Befehle den eigenen Bedürfnissen anzupassen.

Ein Schwerpunkt wird auf die Makroprogrammierung gelegt. Hier finden Sie die Bedienung der Tastaturmakros und eine Programmieranleitung für eigene Makros.

2. Haftungsausschluss

Für die Fehlerlosigkeit der Software und der Dokumentation kann keine Gewähr übernommen werden. Ebenso wird keine Garantie gegeben, dass die Software den spezifischen Anforderungen des Lizenznehmers entspricht. CASO haftet keinesfalls und niemandem gegenüber für kollerale, zufällige, indirekte oder Mängelfolgeschäden, die sich aus dem Kauf oder der Benutzung des NC Edit ergeben.

Die Firma CASO GmbH behält sich das Recht vor, jederzeit den NC Edit zu revidieren oder zu verändern. CASO ist nicht verpflichtet, hiervon irgend jemanden zu unterrichten.

Bei Fragen wenden Sie sich bitte an:

CASO GmbH

Montgelasstr. 13

D-83109 Großkarolinenfeld

Tel.: 08031-356380

Fax: 08031-382558

Email: rosenheim@caso.de

4. Installation

Für die Installation des NC Edit benötigen Sie einen Rechner, der mindestens folgende Anforderungen erfüllt:

- Windows 98/ME/2000/XP
- 32 MB freien Arbeitsspeicher (RAM)
- 20 MB freien Speicherplatz auf der Festplatte

Mit dem Einlegen der mitgelieferten CD-ROM startet die Installationsroutine automatisch. Sie können aber auch das Installationsprogramm von Hand starten. Es heißt „Setup.exe“ und befindet sich im Hauptverzeichnis der CD-ROM. Das Programm kann jederzeit wieder deinstalliert werden. Dazu finden Sie einen entsprechenden Eintrag in der Systemsteuerung unter Software. Während der Installation werden Sie nach der Seriennummer gefragt. Diese finden Sie auf der CD-ROM aufgedruckt.

4.1 Eingabe der Autorisierung

Der NC Edit ist kopiergeschützt. 30 Tage nach der Installation wird beim Starten des NC Edit ein Autorisierungscode verlangt. Diesen können Sie unter der Telefonnummer 08031-356380 erfragen. Bitte halten Sie die angezeigten Informationen bereit. Wenn Sie die Autorisierung vor Ablauf der 30 Tage eingeben möchten, so können Sie das Systemdatum kurzzeitig um einen Monat verändern und dann den NC Edit starten. Nach Eingabe der Autorisierung kann das Systemdatum wieder zurückgestellt werden.

4.2 Netzwerkinstallation

Der NC Edit kann auch zentral auf einem Server installiert werden. Dabei kann jeder Benutzer seine eigenen Konfigurationseinstellungen individuell abspeichern. Die benutzerspezifischen Konfigurationsdateien werden dabei in ein dafür vorgesehenes Verzeichnis geschrieben. Dieses Verzeichnis wird dem

NC Edit durch die Umgebungsvariable NCEDIT_PFAD mitgeteilt. Diese kann unter Windows NT in der Systemsteuerung unter System/Umgebung eingestellt werden. Dazu klicken Sie in das Feld Benutzervariablen für <Benutzer und tragen in das Feld Variable NCEDIT_PFAD ein. In das Feld Wert tragen Sie bitte den Pfad ein, in dem die Konfigurationsdateien abgelegt werden sollen. Bitte vergessen Sie nicht den Set-Knopf zu drücken, damit die Eingaben übernommen werden.

Unter Windows 98 wird die Umgebungsvariable in der Datei Autoexec.bat im Hauptverzeichnis der C:-Festplatte eingetragen. Der Eintrag erfolgt in der Form:

set NCEDIT_PFAD=C:\NCEDIT\USER

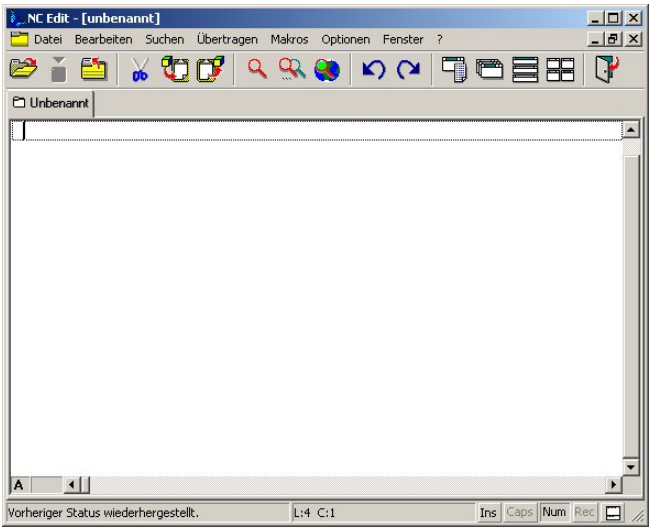
Da unter Windows 95/98 keine benutzerspezifischen Umgebungsvariablen zur Verfügung stehen, bietet der NC Edit die Möglichkeit den Netzwerk-Logonnamen aus der Registry zu lesen und so ein Benutzerverzeichnis anzugeben. Dabei wird die Umgebungsvariable NCEDIT_PFAD auf den Wert NETZ gesetzt. Der NC Edit erzeugt dann die Konfigurationsdateien in einem Unterverzeichnis der NC Edit Installation. Ist z.B. der Benutzer Meier angemeldet, so enthält die Registry unter HKEY_LOCAL_MACHINE\Network\Logon\username den Wert Meier. Der NC Edit speichert dann die Benutzerkonfigurationsdateien im Unterverzeichnis Meier der NC Edit-Installation.

Autorisierungscodes sind nur pro zugreifenden Rechner notwendig. Sie werden zentral in der Datei ncecfg.db im Unterverzeichnis Config der NC Edit-Installation eingetragen. Diese Datei enthält den Abschnitt SET1MODE=0. Die Null ist mit dem Autorisierungscode zu ersetzen. Weitere Autorisierungscodes können in den folgenden Zeilen eingetragen werden.

5. Menüs und Befehle

5.1 Das Hauptfenster

Nach dem Start erscheint folgendes Fenster:



In der Titelleiste wird der Dateiname der momentan aktiven Datei angezeigt. Ist noch keine Datei geladen, so wird ein leeres Fenster geöffnet. Die in den Pull-down Menüs angezeigten Befehle sind auch durch Tastenkombinationen aufzurufen. Diese Tastenzuweisung wird in den Pull-down Menüs neben dem Befehl angezeigt. Die Zuweisung der Tastenkombinationen kann individuell geändert werden (siehe Kapitel Tastenzuweisung). Unterhalb der Pull-down Menüs befindet sich eine Werkzeugleiste. Sie enthält Symbole mit den wichtigsten Befehlen.

Am unteren Rand befindet sich die Statuszeile. In dieser Zeile zeigt der NC Edit Meldungen der ausgeführten Befehle an. Hier wird auch die aktuelle Position des Cursors, sowie der Zustand der Umschalttasten angezeigt.

Die Statuszeile ist ein aktives Element, d.h. Sie können z.B. durch Klicken auf die Zeilen/Spaltenanzeige direkt zu einer bestimmten Zeile springen. Auch die Umschalttasten können hier betätigt werden.

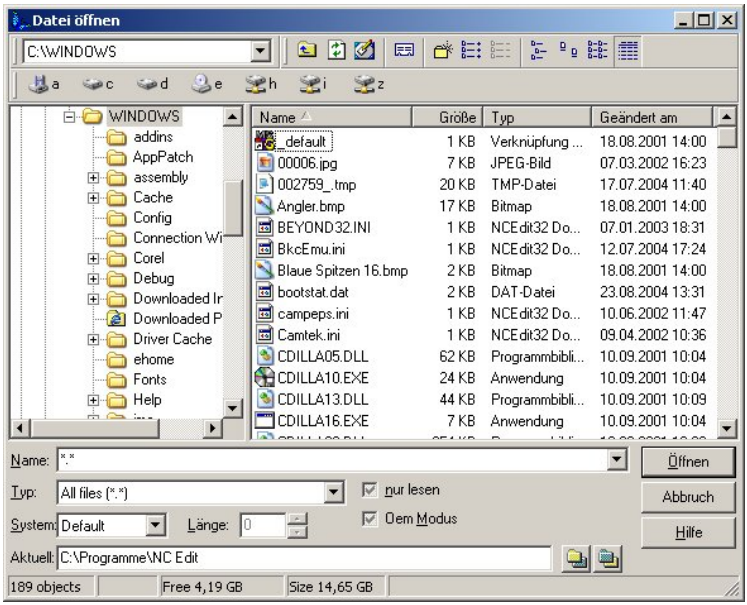
Da das Aussehen des NC Edit weitgehend geändert werden kann, beschreibt das Handbuch den Zustand der Menüs bei der Auslieferung. Menüpunkte können grau hinterlegt sein. Sie sind dann inaktiv, d.h. nicht anwählbar. Das ist dann der Fall, wenn der Menüpunkt zu keinem logischen Ergebnis führen würde, wie z.B. das Speichern wenn keine Datei geladen ist.

Der NC Edit merkt sich Position und Größe des Hauptfensters.

5.2 Das Menü Datei

Im Menüpunkt Datei sind alle Befehle zum Öffnen, Schließen und Drucken von Dateien enthalten.

Mit dem NC Edit können hunderte von Dateien gleichzeitig geöffnet und bearbeitet werden. Der Befehl „Öffnen...“ dient zum Laden einer Datei in ein neues Fenster.
Der Datei-Öffnen Dialog hat folgendes Aussehen:



Es können mehrere Dateien gleichzeitig ausgewählt werden. Dazu muss beim Anklicken des Dateinamens die <Strg>-Taste festgehalten werden. Die Auswahl des OS-Modus legt, fest in welchem Format die Datei geöffnet wird. Der NC Edit erkennt dieses Format selbstständig. Es kann hier aber auch explizit festgelegt werden. Die Recordlänge gibt die Darstellungsbreite der Datei im Binärmodus an.

Bei den Befehlen „Sichern“, „Sichern als“ kann der Dateityp angegeben werden. So kann z.B. eine Datei vom UNIX ins DOS Format geändert werden. Auf diese Weise kann auch z.B. eine LINUX Konfigurationsdatei bequem in der Windows Umgebung erstellt werden und dann im UNIX Format gespeichert werden.

Der Befehl „Information“ gibt Auskunft über die aktuelle Datei. Auch hier kann das Dateiformat geändert werden.

Die Befehle „Schließen“ und „Alle Schließen“ können verwendet werden um Dateien zu schließen (!). Dabei wird geprüft, ob die Datei geändert wurde und ggf. gefragt, ob die Änderungen gespeichert werden sollen.

Mit dem Befehl „Drucker einrichten“ kann das Drucken voreingestellt werden. Hier können die Ränder, Schriftart, Kopf- und Fußzeilen sowie die Druckereinstellungen festgelegt werden. Eine Druckvorschau ist nicht verfügbar. Sie ist für die nächste Version geplant.

5.3 Das Menü Bearbeiten

Mit den Befehlen „Zurück“ und „Zurücklöschen“ können Änderungen einzeln rückgängig gemacht bzw. wiederhergestellt werden. Diese Funktion geht sogar über das Speichern hinaus, d.h. solange die Datei noch geladen ist, können Änderungen auch nach dem Speichern wieder rückgängig gemacht werden.

Die Funktionen „Ausschneiden“, „Kopieren“ und „Einfügen“ entsprechen den Windowsfunktionen der Zwischenablage und beziehen sich auf vorher markierte Textabschnitte.

Mit den Befehlen „Ins Fenster kopieren“, „Verschieben“ kann ein Textabschnitt in dem einen Fenster markiert werden und in ein anderes Fenster an die aktuell Cursorposition kopiert/verschoben werden.

Der Befehl „Markiertes speichern“ setzt voraus, daß ein Textabschnitt mit der Maus markiert wurde. Der markierte Text kann dann mit diesem Befehl in eine Datei geschrieben werden.

Mit dem Befehl Matheoperation können z.B. zwei oder mehrere markierte Zahlen addiert werden. Das Ergebnis befindet sich in der Zwischenablage und kann mit dem Befehl „Bearbeiten“ „Einfügen“ an der aktuellen Cursorposition eingefügt werden. Die vier Grundrechenarten stehen zur Verfügung.

Der Menüpunkt „Markierung halten“ ist eine Umschaltfunktion. Wird er angewählt, so erscheint ein Haken links vor dem Befehl. Eine erneute Anwahl schaltet die Funktion wieder aus – der Haken im Menü verschwindet wieder. Ist der Befehl angewählt, so bleibt ein markierter Text weiterhin markiert, auch wenn Eingaben getätigt werden oder der Cursor mit der Maus positioniert wird.

Der Befehl „Spaltenbereich markieren“ ist eine Funktion, mit der Text auch abschnittsweise senkrecht markiert werden kann. Dies ist z.B. notwendig, um in einem Text in jeder Zeile die ersten fünf Zeichen zu entfernen. Dazu bewegt man den Cursor an den Anfang des zu markierenden Textes, wählt den Befehl an und zieht nun den Cursor bei gedrückter Umschalt(Shift)-Taste über den gewünschten Text. Dieser erscheint nun blau hinterlegt und kann z.B. mit dem Befehl „Bearbeiten/Ausschneiden“ entfernt werden.

Mit „Befehl wiederholen“ kann ein Befehl oder Tastaturmakro mehrfach wiederholt werden. Sollen z.B. 10 Zeichen links vom Cursor gelöscht werden, so wird nach dem Aufruf die Zahl 10 eingegeben und anschließend die Rückwärtstaste gedrückt. Dieser Befehl ist insbesondere bei Tastaturmakros hilfreich. Weitere Informationen dazu finden sie im Kapitel Tastaturmakros.

5.4 Das Menü Suchen

Der NC Edit bietet eine umfangreiche und flexible Suchfunktion. Es können z.B. auch Dateien auf der Festplatte nach einem Suchtext durchsucht werden. Der Suchtext kann sogar mit den von UNIX bekannten regulären Ausdrücken definiert werden. Damit ist es auch möglich nach sonst schwer beschreibbaren Texten zu suchen wie z.B. alle Texte in Anführungszeichen. Weitere Informationen zu regulären Ausdrücken finden Sie im Kapitel reguläre Ausdrücke.

Die Optionen für die Befehle:

Suchen/Suchen und Ersetzen/Suchen in Dateien/Suchen und Ersetzen in Dateien

- Suchstring
hier wird der Text eingegeben, nach dem gesucht werden soll. Schon vorher verwendete Suchtexte lassen sich durch klicken auf den Pfeil auswählen. Wenn ein Textabschnitt markiert ist, so wird er automatisch als Suchstring eingetragen.

- **Wörtlich**
  mit dieser Option wird nach Texten gesucht, die exakt dem Suchtext entsprechen. Reguläre Ausdrücke werden nicht ausgewertet, sondern als Suchtext interpretiert.

- **Regulärer Ausdruck**
  mit dieser Option werden Zeichen wie $%&[* usw. mit einer bestimmten Eigenschaft assoziiert (siehe Kapitel ‚Reguläre Ausdrücke').

- **Zeilenübergreifend**
  diese Option sucht den Suchtext auch über mehrere Zeilen hinweg.

- **Suchdialog offen lassen**
  ist diese Option angewählt, so wird der Suchdialog nicht nach dem Auffinden des Textes geschlossen.

- **Nichts gefunden->schließen**
  wenn der Suchtext nicht gefunden wurde, wird der Suchdialog automatisch geschlossen.

- **Rückwärts suchen**
  die geladene Datei wird nach dem Suchtext ab der Cursorposition in Richtung Dateianfang durchsucht. Ist diese Option nicht angewählt, so ist die Suchrichtung immer auf das Dateiende hin.

- **Globale Suche**
  wenn diese Option angewählt ist, beginnt die Suche am Dateianfang. Anderenfalls wird ab der Cursorposition gesucht.

- **Groß/Kleinschreibung**
  legt fest, ob auf Groß/Kleinschreibung geachtet werden soll.

- **Im markierten Block**
  wenn ein Textabschnitt markiert ist, so kann diese Option die Suche auf diesen beschränken. Diese Option ist nur verfügbar, wenn ein Text markiert wurde.

- **In allen Fenstern suchen**
  legt fest, ob nur das aktive Fenster, oder alle geladenen Dateien durchsucht werden sollen.

- **Gefundenen Text markieren**
  legt fest, ob der gefundene Text markiert werden soll.

- **Spalte**
  hier kann der zu durchsuchende Bereich spaltenweise begrenzt werden.

- **Alias**
  Hinter diesem Knopf verbirgt sich eine veränderbare Liste mit vorgefertigten regulären Ausdrücken.

- **Vorgabe**
  setzt alle Optionen der Suchmaske auf die Vorgabewerte zurück. Die Vorgabewerte sind frei konfigurierbar (-> Kapitel Konfiguration).

- **Alle**
  Dieser Knopf erstellt eine Liste mit den gefundenen Texten. Diese Liste wird unten eingeblendet. Durch klicken auf die jeweilige Zeile kann bequem zu den einzelnen

Fundstellen gesprungen werden. Die Liste kann mit dem Symbol ganz unten Rechts in der Statuszeile ein und ausgeblendet werden.

Der Befehl ‚**passende Klammer finden**‘ kann zu einer öffnenden Klammer die zugehörige schließende Klammer finden (und vice versa). Dazu ist der Cursor entsprechend neben die Klammer zu positionieren. Durch den Aufruf des Befehls springt der Cursor dann zur zugehörigen Klammer. Dies funktioniert auch bei mehrfach geschachtelten Ausdrücken (Kellerspeicher).

Der Befehl ‚**Gehe zur Zeile**‘ öffnet einen Dialog in dem die gewünschte Zielzeile eingegeben werden kann. Dieser Befehl kann auch durch Anklicken der Zeilen/Spalten-Anzeige aufgerufen werden.

5.5 Das Menü Übertragen

Diese Menüpunkte ermöglichen eine Übertragung von Dateien über die serielle Schnittstelle, insbesondere für die Kommunikation mit CNC Werkzeugmaschinen.

- **Parameter**
  öffnet den Dialog zum Festlegen der Übertragungsparameter. Es können bis zu 10 verschiedene Konfigurationen/Maschinen abgespeichert werden. Die Parameterdatei legt die zu ändernde Konfiguration fest. Änderungen bei Schnittstelle oder Sondereinstellungen beziehen sich dann auf diese Konfiguration. Die Optionen der Schnittstelle müssen mit denen der CNC Maschine übereinstimmen. Die Sondereinstellungen im Einzelnen:

- Puffergrößen: sind voreingestellt und sollte nur bei Übertragungsfehlern geändert werden.
- Zeilenende: legt die Kennung von Zeilenende fest. Voreingestellt ist 0D0A für DOS. Einige Maschinen benötigen hier ein 0A (UNIX-Format).

- Hexnullen filtern: filtert alle Hex-Nullen aus dem Datenstrom. Brauchbar um z.B. bei MAHO-Maschinen den Vor- und Nachspann zu filtern.
- Sendenachspann:
  hier kann eine oder mehrere Zeichen nach den Nutzdaten gesendet werden um z.B. der Maschine mitzuteilen, daß die Übertragung beendet ist.
- Empfang beenden: hier wird festgelegt wie das Ende der Datenübertragung von der Maschine zum PC erkannt wird.
- Daten filtern:
  hier kann der Datenstrom auf Nutzdaten innerhalb einer Kennung gefiltert werden; z.B. zwischen den %-Zeichen bei FANUC Steuerungen.

Bei der Einrichtung von Übertragungen hilft ihnen auch gerne unser erfahrenes Team.

Senden

Dieser Befehl fragt nach der gewünschten Konfiguration und sendet die Datei im aktiven Fenster zur Maschine.

Empfangen

Öffnet ggf. ein neues Fenster und startet den Datenempfang von der Maschine.

5.6 Das Menü Makros

Der NC Edit unterscheidet zwischen Tastaturmakros und Makros die in einer der Programmiersprache C ähnlichen Sprache erstellt werden können (siehe Kapitel Makros).
Der Befehle ‚Makro starten', ‚Laden', ‚Liste aller Makros' und ‚Liste aller Globals' sind für die Entwicklung und den Test von C-Makros.

Aufzeichnen Start/Stop startet bzw. beendet den Tastatur-Makrorecorder. Wenn der Recorder gestartet ist, werden alle Eingaben und Befehlsaufrufe aufgezeichnet. Wird der Recorder gestoppt, so erscheint eine Auswahl ob das aufgezeichnete Makro permanent oder nur in der aktuellen Sitzung zur Verfügung stehen soll. Diese Makros können dann bestimmten Tasten zugewiesen werden. Ist der Recorder gestartet, so erscheint das Symbol <Rec> rot hinterlegt in der Statuszeile. Durch klicken auf dieses Symbol kann der Recorder auch gestartet und beendet werden.
Der Menüpunkt ‚**Aufgezeichnete Makros**' öffnet eine Dialogmaske, mit der die Tastaturmakros verwaltet werden können.

5.7 Das Menü Optionen

Der Befehl ‚**Renummerierung**' dient der Neunummerierung von NC Programmen wenn z.B. eine Zeile eingefügt wurde. Das folgende Beispiel erläutert die Funktionsweise:

In das NC Programm:

N1 T6 M6
N2 S3000 M3
N3 G0 X10 Y10 Z20
N4 G1 Z10 F50
N5 M30

Soll eine Zeile eingefügt werden, mit dem Resultat:

N1 T6 M6
N2 S3000 M3

N3 G0 X10 Y10 Z20
N4 G1 Z10 F50
N4 G1 Z5
N5 M30

Die Maschine würde normalerweise die doppelt vorhandene Zeilenkennung N4 nicht verkraften.
Der Befehl Renummerierung mit den Optionen: Startnummer:10, Inkrement:10, Ziffernanzahl 3, Nummernafang:N und Leerzeichen nach der Nummer führt dann zu folgendem Ergebnis:

N010 T6 M6
N020 S3000 M3
N030 G0 X10 Y10 Z20
N040 G1 Z10 F50
N050 G1 Z5
N060 M30

Diese Funktion ist insbesondere bei großen Dateien mit einem Umfang von mehreren Megabyte eine wertvolle Hilfe.

Die ‚**ASCII Tabelle**‘ zeigt die Zuordnung der Zahlen zu den Zeichen in verschiedenen Codierungssystemen an. Vor allem Programmierer, die sich mit Dateiverarbeitung und –konvertierung beschäftigen benötigen diese.

Die ‚**Syntaxüberprüfung**‘ ist eine Rechtschreibkorrektur. Mit ihr können auch unzulässige Ausdrücke in NC Programmen gefunden werden.

Der ‚**Rechner**‘ bietet neben verschiedenen Zahlensystemen auch eine Historie, und Binärfunktionen.

Das ‚**Notizbuch**‘ ermöglicht die Verwaltung einer TODO-Liste. Die Bedienung ist selbsterklärend.

Die ‚**Hexadezimale Darstellung**‘ zeigt die geladene Datei gewohnt im Textmodus an. Zusätzlich zeigt ein zweites Fenster die Datei die Hexadezimale Codierung der Zeichen an.

Die ‚**komprimierte Darstellung**‘ kann zur vereinfachten Darstellung prozeduraler Texte verwendet werden. So können z.B. in einem C-Programm die Funktionsimplementierungen oder in einem NC Programm die Unterprogrammdefinitionen weggeschaltet werden.

Die Beschreibung des Befehls ‚**Konfiguration**‘ finden Sie im Kapitel Konfiguration.

5.7 Das Menü Fenster

Hier finden Sie Befehle zur Anordnung der geladenen Fenster.

5.8 Das Menü Hilfe

Hier können Sie die Online-Hilfe aufrufen bzw. die Seriennummer einsehen.

6. Konfiguration

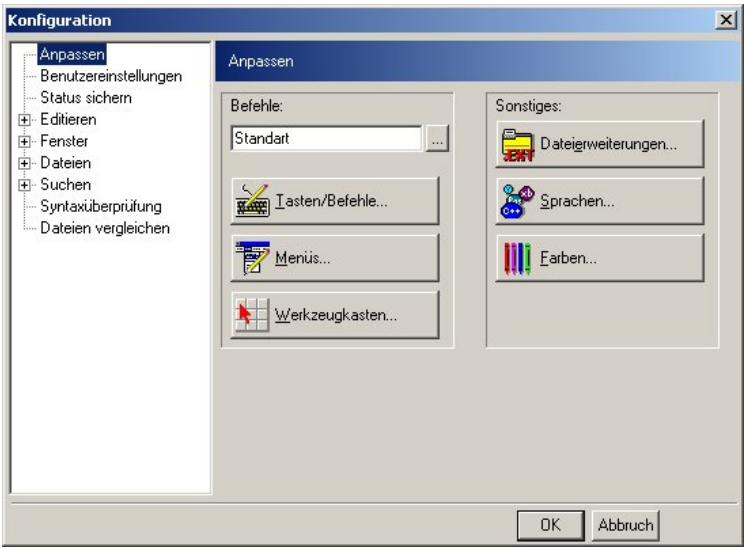
Damit Sie den NC Edit voll ausnutzen können, sollten Sie sich mit den Konfigurationsmöglichkeiten vertraut machen und den NC Edit nach Ihrem Geschmack konfigurieren. Die Konfiguration bei Auslieferung entspricht der allgemeinen Windowsbedienung bzw. den persönlichen Einstellungen des Programmierers.

Die Konfiguration wird mit dem Befehl **Konfiguration** im Menü **Optionen** aufgerufen. Es erscheint eine Maske mit Karteikarten. Im Folgenden werden nur die Optionen beschrieben, die nicht unbedingt selbsterklärend sind. Sollten Sie dennoch nicht die gewünschte Information finden, wenden Sie sich bitte an die Hotline der Fa. CASO Networks (Tel.08031-356380).

Sie können jederzeit die Konfiguration auf den Auslieferungszustand zurücksetzten. Jegliche Änderungen an der Konfiguration werden im Verzeichnis \NCEdit32\config gespeichert. Änderungen an den Tastaturzuweisung speichert der NC Edit in der Datei wcmdmap.db. Alle anderen Änderungen an der Konfiguration werden in der Datei meconfig.db hinterlegt. Wenn Sie diese beiden Dateien löschen, wird wieder die Konfiguration bei Auslieferung verwendet.

6.1 Karteikarte Anpassen

Die Dialogmaske für die Konfiguration wird mit dem Befehl ‚Konfiguration...' im Menü Optionen aufgerufen. Es erscheint folgender Dialog:



Mit dem Knopf ‚**Tasten/Befehle...'** können die Zuweisungen der Befehle zu den Tasten eingesehen und verändert werden. Ein Befehl muß nicht zwingend über eine Taste erreichbar sein. Es ist auch möglich, daß er aus dem Menü aufgerufen wird oder gar keine Verwendung findet. Die Zuweisung kann mit dem Edit-Knopf geändert werden.

Das Menü ist ebenfalls frei editierbar. Der Knopf **‚Menüs'** öffnet den Dialog für das Ändern der Menüs. Das Menü main ist das Menü des NC Edit. Es gibt auch ein weiteres Menü mit dem Namen Context. Dieses Menü erscheint, wenn mit der rechten Maustaste auf den Text geklickt wird. Mit Edit kann das Fenster zum Ändern des Menüs geöffnet werden. Hier findet man die einzelnen Menüpunkte abgebildet. Wird nun erneut der Edit-Knopf gedrückt, so erscheinen die Einstellun-

gen des ausgewählten Menüpunktes. Hierbei entspricht der Menütext dem Text, der im Menü angezeigt wird. Der Level legt die Menüebene fest, d.h. ob der Menüpunkt einem Untermenü zugeordnet ist oder nicht. Command legt den Befehl fest, den der Menüpunkt aufrufen soll. Dieser kann durch Klicken auf den Knopf mit den drei Punkten auch aus dem Tastenzuweisungsmenü ausgewählt werden. Nun liegen mehrere Dialoge übereinander. Sie müssen der Reihe nach mit Schließen geschlossen werden.

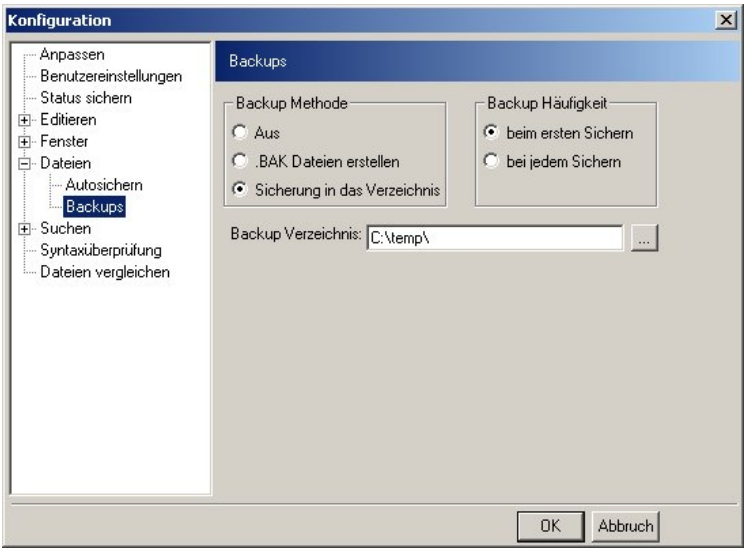
Mit den **‚Dateiendungen‘** kann das Syntaxhighlighting eingestellt werden. Der NC Edit kann abhängig von der Dateiendung den Text verschiedenfarbig einfärben. So werden in einem NC Programm z.B. alle G- und M-Befehle fett dargestellt. Es ist z.B. auch möglich alle Z-Achsen-Bewegungen in rot darzustellen. Für die gängigsten Dateiendungen sind bereits vorgefertigte Einstellungen vorhanden. Wenn eine Datei mit unbekannter Endung geladen wird, so verwendet der NC Edit eine sinnvolle Vorgabe. Mit dem Edit-Knopf können Dateiendungen zum Ändern ausgewählt werden. Jeder Dateiendung ist eine Sprache (über den Knopf Language) zugeordnet. In der Sprache sind die reservierten Worte für die Farbdarstellung eingetragen. Eine Sprache kann auch mehreren Dateiendungen zugeteilt sein. Wenn Sie eine Farbdarstellung für eine Sprache und damit auch für eine Dateiendung erstellen wollen, so ist es am einfachsten die Bestehenden zu studieren.

6.2 Automatisches Sichern, Backups

In der Karteikarte Dateien im Konfigurationsdialog finden Sie die Optionen zu automatischem Sichern und dem Anlegen von Sicherungsdateien. Unter ‚**Autosichern**‘ können nach einem einstellbaren Zeitintervall oder nach einer einstellbaren Zeit ohne Benutzereingaben die geöffneten Dateien automatisch gesichert werden.

Ob der NC Edit vor dem Sichern Kopien der Dateien anlegt oder nicht, kann im Karteikasten Backups eingestellt werden. Es können entweder Sicherungen mit der Dateiendung .BAK erstellt werden oder eine Kopie im einem einstellbaren Verzeichnis erstellt werden. Die Backup-Häufigkeit legt fest, ob

nur jeweils der letzte Stand der Datei gesichert wird oder ob eine komplette Historie geführt werden soll.



6.3 Status sichern

Im Auslieferungszustand wird der NC Edit immer mit einem leeren Fenster starten. Der NC Edit kann aber auch so konfiguriert werden, daß er beim nächsten Start den Zustand der letzten Sitzung wiederherstellt. Es werden dann alle Dateien geladen, die beim Verlassen des NC Edit geöffnet waren. Auch alle anderen Listen werden mitgeführt, wie z.B. Suchergebnisse oder Historyeinträge der Dialoge.

Diese Option wird im Konfigurationsdialog im Karteikasten ‚**Status sichern**‘ eingestellt. Sie können eine globale Statusdatei in einem einstellbaren Verzeichnis wählen oder die Statusinformationen für jedes Verzeichnis getrennt einstellen. Wenn kein Pfad für die globale Statusdatei angegeben wurde, so wird sie im Config-Verzeichnis bzw. im Benutzerkonfigurationsverzeichnis abgelegt.

7. Reguläre Ausdrücke

Reguläre Ausdrücke sind so etwas wie Platzhalter bzw. Wildcards. Sie werden hauptsächlich zum Suchen verwendet.

Was wenn Sie z.B. nach Zeilen suchen wollen, die mit dem Wort ‚wichtig' beginnen?

Übliche Suchfunktionen liefern hier nur **alle** Zeilen in denen das Wort ‚wichtig' vorkommt.

Mit diesen Ausdrücken kann man auch so etwas wie 'Anfang einer Zeile' oder 'alles was zwischen Anführungszeichen steht', beschreiben.

Reguläre Ausdrücke sind also Zeichen mit besonderer Bedeutung. So z.B. repräsentiert das Zeichen % den Anfang einer Zeile.

Reguläre Zeichen sind:

SUCHEN

*	0 oder mehr Funde des vorherigen Zeichens oder Ausdrucks
+	1 oder mehr Funde des vorherigen Zeichens oder Ausdrucks
?	beliebiges Zeichen
[]	eine Klasse von Zeichen
[~]	eine Klasse von Zeichen außer denen in der Klammer
%	Anfang einer Zeile
$	Ende einer Zeile
@	trifft (oder ersetzt) nächstes Zeichen wörtlich
{}	schließt eine Gruppe von 1 oder mehreren Ausdrücken ein
|	vorheriges oder nächstes Zeichen oder Ausdruck

ERSETZEN

$	Fügt eine Zeilenschaltung ein
%	Löscht ein Zeichen
&	Fügt den original gefundenen Text ein
^	Plaziert den Cursor an dieser Position im ersetzten Text
#n	Fügt Text ein, passend zur Gruppe Nummer "n"

7.1 Beispiel: Suchen nach Klammerausdrücken

Folgendes Beispiel zeigt das Suchen nach Texten, die zwischen Klammern stehen. Der reguläre Ausdruck hierfür lautet:

(?*)

Es soll also nach einer öffnenden Klammer gesucht werden, gefolgt von einem beliebigen Zeichen, das hier mit dem Fragezeichen repräsentiert wird. Der Stern besagt, daß dieses Zeichen beliebig oft wiederholt werden kann. Nach diesen Zeichen muß eine schließende Klammer folgen. Somit werden alle Klammern gefunden, die keine oder einen beliebig langen Text einschließen.

7.2 Beispiel: Suchen nach Anführungszeichen

Folgendes Beispiel sucht nach zusammengehörenden Anführungszeichen. In den meisten Programmiersprachen werden Texte, sogenannte Strings, durch Anführungszeichen vor und nach dem Text klassifiziert. Der reguläre Ausdruck:

{"?*"}|{'?*'}

Sucht nach Texten, die entweder zwischen zwei Anführungszeichen (") oder zwischen zwei Hochkomma (') stehen.

Hier wird wie im vorherigen Beispiel nach einem Text zwischen zwei Zeichen gesucht. Um aber gleichzeitig beide Begrenzungszeichen anzugeben ist es notwendig, beide Suchausdrücke anzugeben. Die Suchausdrücke werden dann mit dem senkrechten Strich |, auch Pipe genannt (zu erreichen z.b.

über die Eingabe , getrennt. Dieser senkrechte Strich entspricht einem logischen ODER.

Es wäre also auch möglich den Suchausdruck mit "|‘?*“|‘ zu formulieren. Hier würde nach einem Text gesucht, der mit " oder ' beginnt, ein oder beliebig viele Zeichen enthält und mit " oder ' endet. Dabei werden aber auch Texte gefunden, die z.B. mit einem Hochkomma beginnen aber mit einem Anführungszeichen enden. Um dies auszuschließen kann wie oben gezeigt, die Zugehörigkeit mit geschweiften Klammern klassifiziert werden.

7.3 Beispiel: Leerzeilen Suchen und Löschen

Folgendes Beispiel zeigt das Suchen nach einer Leerzeile. Es wird auch ein Ausdruck zum Löschen dieser Leerzeile formuliert. Der reguläre Ausdruck:

Suchen: %$

Ersetzen: %

Im Suchstring stellt das Prozentzeichen eine Zeilenanfang dar. Das Dollarzeichen repräsentiert das Zeilenende. Ein Zeilenende direkt nach dem Zeilenanfang definiert also eine Leerzeile.

Für das Ersetzen hat das Prozentzeichen eine andere Bedeutung. Hier bewirkt es das Löschen des gefundenen Suchstrings.

7.4 Beispiel Parametertausch

Dieses Beispiel zeigt die Addressierbarkeit von Suchgruppen. Bei allen Aufrufen der Funktion gotoxy(x,y) sollen die Parameter vertauscht werden. Dazu der reguläre Ausdruck:

Suchen: gotoxy({?+},{?+});

Ersetzen: gotoxy(#1,#0);

Die Ausdrücke im Ersetzungsstring #0 und #1 bezeichnen die gefundenen Texte aufgrund der in geschweiften Klammern angegebenen Suchklassen. Die Suchklassen werden dabei von links nach rechts - bei Null beginnend - durchnummeriert.

8. Makros

8.1 Tastaturmakros

Der NC Edit ist in der Lage die von Ihnen getätigten Tastatureingaben und Menüaufrufe aufzuzeichnen. Diese aufgezeichneten Makros können einer freien Taste zugewiesen werden.

Die Tastaturmakros sind ein sehr leistungsfähiges Hilfsmittel. Es hat sich gezeigt, daß der Nutzen dieser Makros erst erkannt wird, wenn sie ausprobiert wurden.

Unterschieden werden temporäre Makros und permanente Makros. Die temporären Makros werden nur in der aktuellen Sitzung des NC Edit gespeichert. Die permanenten Makros sind auch bei späteren Aufrufen des NC Edit verfügbar. Ein aufgezeichnetes, temporäres Makro kann in ein permanentes umgewandelt werden.

Zum Starten des Makrorekorders dient der Befehl ‚Aufzeichnen Start/Stop' im Makromenü. In der Statusleiste erscheint das Symbol REC rot hinterlegt, wenn der Rekorder läuft. Anderenfalls ist es grau hinterlegt. Der Rekorder kann auch durch Klicken auf das Symbol gestartet und gestoppt werden.

Nach dem Stoppen des Rekorders erscheint ein Dialog zur Auswahl des Makrotyps. Hinweis: Wurden während der Aufzeichnungsphase keine Tastatureingaben getätigt, so erscheint dieser Dialog nicht; das Makro wird verworfen, da es leer ist.

Mit dem Menüpunkt ‚aufgezeichnete Makros...' wird der Dialog zur Makroverwaltung aufgerufen. Nur wenn schon ein temporäres Makro aufgezeichnet wurde, zeigt dieser Dialog den Knopf ‚Temporäres Makro sichern'. Dieser Knopf wandelt temporäre Makros in permanente. Der Dialog zur Makroverwaltung ermöglicht das Löschen, Kopieren und Bearbeiten von Tastaturmakros.

8.2 Makros mit dem Compiler erstellen

Der NC Edit hat einen Makrocompiler integriert. Dieser Compiler übersetzt einen Quelltext in ein ausführbares Makro. Die meisten Befehle im NC Edit wurden mit diesem Compiler erstellt. Der Quelltext muß in der passenden Form für den Compiler formuliert werden. Diese Form oder besser Programmiersprache ist stark an die Programmiersprache C angelehnt. Sie enthält aber auch Elemente aus den Sprachen Pascal, Modula2 und BASIC. Im Nachfolgenden finden Sie Programmierbeispiele und eine Referenz der Sprache.

Das folgende, einfache Beispiel zeigt ein Makro zum Löschen der aktuellen Zeile. Der Nutzwert des Makros ist sehr gering, da der NC Edit schon ein Makro zum Löschen einer Zeile enthält(standardmäßig auf der Taste <Strg>-Y). Es soll dem grundsätzliche Aufbau zeigen.

Der folgende Quelltext wird in eine Datei zeile.s geschrieben. Die Endung .s kennzeichnet den Dateityp als Makrodatei.

```
macro_file Zeileloeschen;

macro Zeileloeschen {

/************************************************

Name: Zeileloeschen

Beschreibung: Löscht die aktuelle Zeile

************************************************/

 del_line;

}
```

Die erste Zeile legt den Dateinamen der Makrodatei fest. Eine Makrodatei kann mehrere Makros beinhalten. Der Makroquelltext wird zwischen geschweifte Klammern geschrieben. Jeder Befehl wird mit einem Semikolon abgeschlossen. Ein Kommentar wird zwischen die Zeichen /* und */ geschrieben. Kommentare werden vom Compiler ignoriert. Dieses Makro benutzt eine einzige Bibliotheksfunktion: del_line.

Diese Datei wird nun geladen und mit der Taste F9 compiliert. Dadurch erstellt der Compiler eine Zeilelo.mac im Verzeichnis \ncedit32\mac.

Um das Makro aufzurufen kann der Menüpunkt „Makro starten“ im Makromenü benutzt werden. Dies ist natürlich auf Dauer etwas unbequem. Deshalb legen wir den Befehl im Menü ab.

Dazu muß der neue Befehl im Menü ‚Optionen/Konfiguration/Tasten‘ eingefügt werden. Nach dem Aufruf des Einfügen-Knopfes müssen noch Name und der Befehl eingegeben werden. Hier die Beschreibung der einzelnen Felder:

Name: Beschreibt den Befehl

Command Line: Zeileloeschen

Type: Makro

Mit Primary Key und Secondary Key können Tasten für diesen Befehl vergeben werden. Unter Button kann der Text und das Symbol für die Darstellung im Werkzeugkasten eingestellt werden. Nun können alle Fenster wieder mit Schließen geschlossen werden.

Um nun diesen Befehl in das Menü einzufügen sind folgende Schritte notwendig:

Unter ‚Optionen/Konfiguration/Menüs‘ wird das Menü main editiert. Es erscheint ein Fenster mit einem Abbild des NC Edit Menüs. Nun ist der Balken auf das Menü Bearbeiten zu positionieren (unterhalb von ‚Einfügen‘) und der Knopf einfügen zu drücken. Unter Menütext wird nun der Befehl eingetragen, so wie er im Menü erscheinen soll, also z.B. ‚Zeile löschen‘. Der Level gibt die Menüebene an und sollte 2 betragen. Unter Command wird nun der Befehl ‚Zeileloschen‘ eingetragen. Durch Klicken auf den Knopf mit den drei Punkten kann der Befehl auch aus den Tastenzuweisungen ausgewählt werden. Wenn wieder alle Fenster geschlossen wurden, steht der neue Menüpunkt zur Verfügung.

8.2.1 Struktur der Compilersprache

Die Makrosprache des NC Edit unterscheidet keine Groß-Kleinschreibung. Folgende Strukturen entsprechen den Strukturen der Programmiersprache C:

```
Do (Statement) .while (expression);
for ( init-expression; test-expression; increment-expression )
      statement;
while ( expression ) statement;
if ( expression )  statement;
if ( expression 1 )  statement 1;
 else if ( expression 2 ) statement 2;
else statement 3;
switch ( integer_expression or string expression )
{case n : statement 1;
          break;
  default : statement 2;
          break;
}
```

Auch struct und Arrays entsprechen der C-Syntax.

Es werden folgende booleanschen Ausdrücke unterstützt:

x == y	x gleich y
x != y	x ungleich y
x < y	x kleiner y
x > y	x größer y
x <= y	x kleiner gleich y
x >= y	x größer gleich y
x && y	x logisch UND y
x & y	x bit-wise UND y
x || y	x logisch ODER y
x | y	x bit-wise ODER y

Es gibt die Datentypen Char, int, string, real.
Strings können im Gegensatz zu C mit dem +-Operator konkatiniert werden. Auch die sonst nur für Integer oder Real zur Verfügung stehenden Vergleichsoperatoren (<,>,<=,>=,==,!=) sind verfügbar. Kommentare werden wie in C mit den Zeichen /* und */ bzw. zeilenweise mit // gekennzeichnet.

8.2.2 Befehlsreferenz des Compilers

8.2.2.1 Block Operationen

BLOCK_BEGIN (void function)
Markiert den Anfang eines Blockes

BLOCK_END (void function)
Markiert das Ende eines Blockes

BLOCK_OFF (void function)
Schaltet die Blockmarkierung aus

BLOCK_STAT (integer function)
Rückgabewert 0: es ist kein Block markiert
Rückgabewert 1: es ist ein Block markiert

Copy_block (void function)
kopiert den markierten Text an die aktuelle Cursorposition

move_block (void function)
verschiebt den markierten Text an die aktuelle Cursorposition

Delete_block (void fuction)
löscht den markierten Block

8.2.2.2 Cursorpositionierung

at_eof (integer function)
liefert true (1) wenn der Cursor hinter dem Dateiende steht

at_eol (integer function)
liefert true (1) wenn der Cursor am Zeilenende steht

c_col (integer function)
liefert die Spalte der aktuellen Cursorposition

c_row (integer fuction)
liefert die Reihe der aktuellen Cursorposition

up (void function)
Positioniert den Cursor eine Zeile höher

down (void function)
Positioniert den Cursor eine Zeile tiefer

left (void function)
Positioniert den Cursor ein Zeichen nach links

right (void function)
Positioniert den Cursor ein Zeichen nach rechts

EOF (void function)
springt ans Dateiende

TOF (void function)
springt an den Dateianfang

EOL (void function)
springt ans Zeilenende

Goto_col (integer col_num)
springt zur Zeile col_num

Goto_row (integer col_row)
springt zur Spalte row_num

8.2.2.3 Stringmanipulationen

ASCII (char)
liefert den ASCII-Wert des Zeichens z.B. jx=ASCII(‚A‘);

CHAR (integer)
liefert das Zeichen zu dem übergebenen Wert

Copy (string, pos, num)
liefert einen Teilstring von string.
z.B. Temp_Str = Copy(Temp_Str, 1, 3); die ersten drei Zeichen von Temp_str

STR (integer)
wandelt ein integer in einen String

Val (string)
wandelt einen String in einen Integer

LENGHT (string)
liefert die Länge des Strings

XPOS(substring, string, start_position)
liefert die Position des substring in string ab der Suchposition start_position.